A human-centred model for network flow analysis

Thibaud Merien[†], David Brosset[†] Xavier Bellekens [*] Christophe Claramunt[‡]
[†] Naval Academy Research Institute, Chair of Naval Cyber Defence, Lanveoc, France
[*]Division of Cyber Security, Abertay University, Dundee Scotland
[‡]Naval Academy Research Institute, Lanveoc, France

***Abstract*—Computer networks are ubiquitous and growing exponentially, with a predicted 50 billion devices connected by 2050. This tremendous growth dramatically increases the attack surface of both private and public networks. These attacks often influence the behaviour of the system, leading to the detection of the attack. In this manuscript we model the path of an attack through the network by graphs. The model developed aims to better integer attackers intentions. Using the data produced by 5 honeypots, we apply our model. The preliminary results show that the approach is useful to rapidly detect anomalies in the experiment dataset.**

I. Introduction

Computer networks are ubiquitous and play a pivotal role in the way users and machines interact with each other. The advent of the Internet of Things (IoT) has significantly increased the number of devices connected to the Internet, with an estimation of 50 billion devices connected by 2020, hence increasing both the network complexity and attack surfaces on both networks and devices.

In the field of telecommunications, a perfect network is a set of interconnected nodes. On the Internet, however, a system newly connected to the Internet is likely to be contacted by thousands of malicious nodes, mostly bots, from all around the World in less than 24 hours, with the main purpose of probing the newly connected system.

In order to protect networks, it is thus necessary to understand the motivations of the attackers, the methods applied, but most importantly, its identity. This can be achieved through profiling the attacks, hence being able to both understand and learn about the actions of an attacker. This can be achieved by modelling the attack flow path within a network.

Network modelling is complex due to the number of paths network packets can follow. In this paper a model based on graph theory is developed. This model aims at representing network activities and to categorize network users as either the author or the victim of a cyber-attack.

To the best knowledge of the authors the majority of models disregards the users, then this model tends to include the user has a central piece of the model. The main contribution of this paper is the introduction of a new model centred around the user. The model helps refine malicious users and attack characteristics.
Furthermore, numerous graphs derived from the model highlights peculiarities in the network traffic, with the ability to identify patterns that can be used to understand the intentions of a malicious user as well as detect misconfigurations and services failure.

The remainder of this paper is organized as follows. Section II presents the state-of-the-art in modelling cyber-attacks and cyber-security visualization. Section III introduces the main model as well as different applications that generate new representations of network communications. Section IV presents a case study realized on five different honeypots scattered around the world, validating our approach, finally, Section V concludes this paper.

II. Related Works

Network activity models have different goals, one of them is to improve understanding of cyber-attacks as well as increase the detection rate and speed. This method allows improving the cyber-defence and provide better tools to monitor networks and systems [7]. Generally, humans are particularly good at recognizing visual patterns, therefore, transforming data and numbers into figures (e.g. plots, graphs, etc.) and images can substantially improve cyber-attack detection and understanding through visualization of important information. The domain of security visualization, has recently been expanded into Human Computer Interaction (HCI) [14, 26, 27]. Security visualization can be classified into three types according to [31]: geographic visualizations, abstract topological representations and plot-based representations. Each type is based on different models in order to define the core elements used to produce visual results.

Different approaches and techniques have been used such as graph theory or Petri Nets [11, 29, 30] to model networks. Graph theory is one of the favourite models used to represent activities in space and time. Such modelling approach is often used in GIS for example to model urban spaces [12, 13]. However, NetFlow models are one of the most used methods to deal with network information [15, 32]. NetFlow is the aggregation of packets based on information such as the source and destination IP address, IP protocol, source and destination ports as well as the type of service. This method is used to help monitor networks rapidly but it is often not powerful enough to detect anomalies or outliers.

Each model is based on a formal representation in order to use the abstraction level and the operators offered by the theory chosen. When identifying cyber-attacks, attack trees and attack forests are often preferred [3, 19]. In an attack tree the root node represents the final goal of the attacker while the sub-nodes are the steps leading towards the goal. This structure is also used to build cyber-defence trees or countermeasures [21]. Depending on the model chosen, different patterns can emerge. These patterns can then be used for the attribution of cyber-attacks, find the identity and/or the location of an attacker [5, 18, 23, 28] and are generally used for cyber-threat intelligence.

Note, however, that the attribution of cyber-attacks is a very challenging problem as numerous tools and software enable hackers to remain anonymous.

To hide their identity, attackers can use Virtual Private Networks (VPN), proxy servers or the TOR network. These techniques were primarily developed to avoid censorship in non-democratic countries and are now primarily used for criminal activities [16]. The principle behind all these methods remains the same: it enables a connection to a server (centralized or decentralized) that hides the true identity of a user. However, if knowing the real identity of the attackers is often impossible, anonymous servers are known (i.e. blacklist) and alerts can be triggered when their use is detected.

Another way to become anonymous on the Internet is to use compromised computers through a botnet. The compromised computers called bots, zombies or ghost computers, are controlled remotely and are used to attack other computers or servers. Botnets have the ability to launch large scale attacks such as Distributed Denial of Service (DDoS) against major services providers, where the chance of success of the attack is strongly correlated with the number of computers attacking. Botnet detection is a very dynamic research field, where the objective is to detect abnormal network communications in order to identify whether a computer is compromised or not. Several methods have been designed to detect bots either by analysing network behaviour [25], or through clustering techniques [8]. Criminal activities heavily rely on the anonymity factor to perpetrate infractions. However, anonymity is only one component as highlighted in criminology [9], other factors play an important role when perpetrating crime.

Numerous crime prevention frameworks have been proposed in criminology and sociology, focusing on different aspect of the crimes committed, in order to provide an insight on crime trends. These models often consider the offender as the main factor, leaving out important components such as the guardian, or a suitable target, hence for the purpose of this manuscript the use of Routine Activity Theory (RAT) has been privileged [4, 20, 22].

RAT assesses the conditions needed from crime to take place in order to explain Direct Contact Predatory Crime (DCPC), by including spatial and temporal patterns alongside situational awareness. RAT is described by the three following components.

- Absence of Capable Guardian
- Suitable Target
- Motivated Offender

RAT implies that crime occurs due the lack of a capable guardian (i.e. Network Firewall, Access Control Lists, Antivirus, etc.), the presence of a suitable target (i.e. a weak host on a network, a user vulnerable to social engineering attacks, etc.) and the presence of a motivated offender (i.e. a malicious user, a script kiddie, ...).

Human actions are at the center of the modelling approach presented. Placing the human in the attack process allows to map the intentions, desires of the user as well as include and analyse all the components provided by the RAT principle.

III. Modelling approach

This section describes the elements of our model and their interconnections. First, we present the main principles of the model, based on graph theory, then we introduce node fusion rules to analyse network activities and understand what is going on during the attack.

A. Main Principles

The objective is to model the network activity produced by a complex system composed of many sub-systems. This model can then further be used for the analysis of the activity qualifying normal and abnormal behaviours by identifying intention and desires.

An attacker (h_{atk}) who wants to compromise a target system, (h_{trg}), performs an action to reach his objective. According to the theory of action, these actions are related to intentions and desires [2]. Therefore, the attacker and the victim are connected by the attacker intention through their own systems as shown in Figure 1.

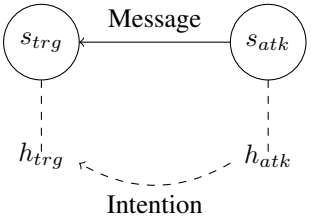

Fig. 1. Network model with intentions

However, numerous systems can co-exist between the attacker and the victim. It is nearly impossible to know the exact path of a message on a network. To this end, a black box is adopted to represent unknown systems used to transmit packets. Figure 2 highlights the different components of a network including the attacker and the victim. Note that, the number of systems included in the black box is unknown.

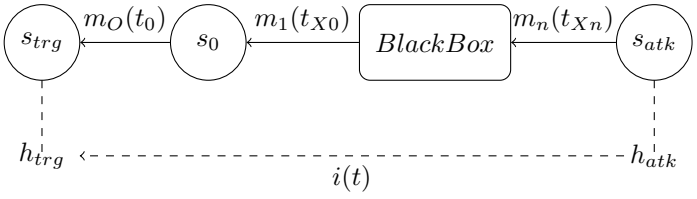

Fig. 2. Network Model with humans

Let S be the set of systems and M the set of network messages. The path between the victim system (s_{trg}) and the attacker system (s_{atk}) can be formally defined using graph theory. As shown in Equation 1, $G_{atk}(t) = (S, M)$ represents the path composed by the systems $\in S$ used by the attacker (h_{atk}) to transmit a message $\in M$ to the victim h_{trg} at a time t.

$$G_{trg,atk}(t) = s_{trg}, m_0(t), s_0, BB[t_0, t_{n-1}], m_n(t_n), s_{atk} \quad (1)$$

Where $m(t) \in M$ represent messages between systems, $s_0 \in S$ the system is directly connected to the victim's system $s_{trg} \in S$. BB, and where the black box, is a set of systems and unknown messages.

A black box can be defined by Equation 2 :

$$BB_{trg,atk}(t) = [s_1, m_1(t), ..., s_n m_n(t_n)] \tag{2}$$

For a period of time (t_0, t_n), $G_{trg,atk}(t_0, t_n)$ represent the graph composed of all the paths between an attacker ($\in A$, the set of attackers) and a victim (trg), as shown in Equation 3.

$$G_{trg,atk}(t_0, t_n) = \prod_{A} G_{trg,atk_A}(t_i) \mid t \in [t_0, t_n] \tag{3}$$

For a period of time (t_0, t_n), the black box is defined by:

$$BB_{trg,atk}[t_0, t_{n-1}] = [s_1, m_1(t_1), ..., s_n m_n(t_n)] \tag{4}$$

As the black box and the system of the attacker are unknown, for the rest of the paper we define $G_{trg,atk}(t)$ by Equation 5.

$$G_{trg,atk}(t) = s_{trg}, m_0(t), s_0^{atk} \tag{5}$$

Nodes are defined by the IP address and edges are messages *i.e.* network packets. Using the IP address of the nodes, we can obtain several information about a system, such as its geolocation. From messages between systems we can extract the original intention of the attacker.

We define the functions $ipv4$, $country(s)$, $city(s)$, $coord(s)$, $s24(s)$, $s16(s)$ returning, the IP address, the country, the city, the coordinate, the /24 subnet and the /16 subnet of a system $s \in S$ respectively. The function $blacklisted(s)$ returns 'true' if an IP address is included in a known lists of IP addresses that are considered suspicious, and 'false' otherwise. The function $intent(m)$ returns the intention of the attacker using the content of the message $m \in M$.

The model presented allows for an analysis of network traffic and intentions that can be used to detect abnormal traffic. Furthermore, by applying fusion rules using node information, it is possible to merge nodes and create clusters from nodes and edges information.

B. Graph Simplification

The different paths of the aforementioned model are composed of numerous nodes and edges over a period of time. In order to highlight patterns contained in these paths, a simplification process is applied. This process can be seen as flow aggregation similar to *C-flow*, introduced by [8]. *C-flow* is a packet aggregator over a period of time. In order to be aggregated, packets must share a protocol (*e.g.* TCP or UDP), a source IP, a destination IP and port numbers.

In our approach the simplification is built by the aggregation of packets over a period of time, however, our model takes the intention of the attackers into account.The Nodes and edges information can be used to merge system nodes, ultimately, producing new graphs.

The different functions used for the aggregation were defined previously in Section III, *i.e.* ($ipv4(s)$, $country(s)$, $city(s)$, $coord(s)$, $s24(s)$, $s16(s)$). According to Rule 1, nodes must share IP addresses and be connected to the victim by messages carrying the same intention in order to be merged together.

Equation 6 and Equation 7 represent an attack against a single victim, by two attackers. Let $atk_1 = G_{trg,a_1}(t_0, t_n)$ and $atk_2 = G_{trg,a_2}(t_0, t_n)$ be two graphs representing all the paths between a victim (trg) and two attackers (a_1 and a_2).

$$G_{trg,a_1}(t_0, t_n) = [s_{trg}, m_0^{a_1}(t), s_0^{a_1}] \mid t \in t_0, t_n \tag{6}$$

$$G_{trg,a_2}(t_0, t_n) = [s_{trg}, m_0^{a_2}(t), s_0^{a_2}] \mid t \in [t_0, t_n] \tag{7}$$

The similarity between the two systems $s_0^{a_1}, s_0^{a_2}$ is identified by applying the similarity rule defined by Rule 1.

Rule 1 (Nodes Similarity)**.**
$\forall t_i, t_j \in [t_0, t_n] \mid t_j = t_i + \Delta t$

$$\bigwedge_{s_0^{a_1} \in G_{trg,a_1}(t_0,t_n), s_0^{a_2} \in G_{trg,a_2}(t_0,t_n)} \bigwedge_{\exists m_0^{a_1}, m_0^{a_2} \mid (s_{trg}, m_0^{a_1}(t), s_0^{a_1}), (s_{trg}, m_0^{a_2}(t), s_0^{a_2})} \left(\begin{bmatrix} ipv4(s_0^{a_1}) = ipv4(s_0^{a_2}) \\ \wedge \\ intent(m_0^{a_1}) = intent(m_0^{a_2}) \end{bmatrix} \Rightarrow Sim(s_0^{a_1}, s_0^{a_2}) \right)$$

Where Δt represent the period of time considered building the aggregation of messages.

For example, let's consider the graph $G_x[T]$ composed by all messages received by x from a set of attackers over a period of time T. The simplification process using Rule 1 with $\Delta t =$ 10 minutes returns a new graph $G_x^{fusion}[T]$. This graph is composed of all the nodes included in the original graph $G_x[T]$ sharing the same IP address as well as all the messages with the same intention.

IV. Case Study

In order to validate the model presented, a large dataset of network traffic is necessary. The number of publicly available dataset is scarce, due to the lack of metadata, and information related to the generation of the dataset. To this end, and for the purpose of this manuscript we generated a dataset using honeypots.

A honeypot is a security tool that can be analysed, probed, attacked and compromised without risk for a network infrastructure [24]. Honeypots are often used to deceive attackers, study attacking methods as well as obtain new and current malware samples [1, 6, 17].

Various types of honeypots exist, each dedicated to a specific use. Honeypots can be classified into three categories: low interaction honeypots, medium interaction honeypots and high interaction honeypots. Low interaction honeypots have limited interactions between the system and the attackers. The honeypot only emulate services. This type of honeypot

presents a low level of risk due to the low interaction. Medium interaction honeypots are between low level interaction honeypots and high interaction honeypots. While the honeypot is deprived of an operating system, it emulates complex services enabling interaction with malicious users. Finally, high interaction honeypots are the most complex type of honeypots. These run a full operating system including services and a complex configuration. The main advantage of high interaction honeypots is that services are not emulated, hence, attackers are interacting with a real target while the owners can track the attackers by capturing all their interactions.

A. Experiment

For the purpose of this research a high interaction honeynet was set up, gathering detailed information on a controlled infrastructure. In order to correlate data from multiple sources, 5 virtual private servers were purchased in 5 different locations (Fremont, Newark, London, Tokyo and Singapore). Figure 3 shows the five location across the world.

Each honeypot was deployed with the same configuration. All of them running Ubuntu 16.04 LTS with an SSH server (openSSH), an FTP server (Pure-ftpd) and a web server (Apache2). The web server hosted an authentication page with a PHP script to track login attempts. All network messages captured were stored in a daily PCAP file using the TCPDUMP command. 10 Gb of raw data was generated within a month.

Applying the RAT theory principles, a capable guardian was omitted, *i.e*. no firewall was configured and the services running on servers are the ones which are often the most targeted by attackers.

B. Data analysis

In this section the data gathered over a period of one month by the honeynet is analysed.

For this experiment we classified network messages into 7 different intentions, divided into 2 main categories: *Information gathering* and *attacks*. In the information gathering category, 4 classes are defined, indicating the main characteristics of the targeted network: Network infrastructure, DNS (Domain Name Server), ICS (Industrial Control Systems) and Web. The attack category concerns the secure remote access attempts and unsecured remote access attempts.

Table I defines how the protocols have been used to classify intentions. The four intentions defined by the *information gathering* category are related to the first phase of the cyberkill chain, the reconnaissance phase, while the intentions defined by the *attack* category, refers to post-reconnaissance. The intention classification is related to the objectives of the analysis and the case study. It is important to note that the intentions used in this work may not be suited to another case study, for example, for a local area network legitimate intentions or pre-approved intentions might be considered.

The characteristics of the honeypots and their location in the world are provided in Table II. The first row represents the number of distinct IP addresses that established a connexion with the honeypots. The second row represents the number of network packets exchanged between the honeypot and the attackers. The third and fourth rows indicate the number of protocols used against each honeypot respectively. The last row of Table II indicates the number of countries the attackers operated from, by geolocating IP addresses. Wireshark does not offer the ability to directly assign a geolocation to an IP address using the GeoIP API. This API is based on a free database, GeoLite[1].

TABLE I. PROTOCOL ATTRIBUTION FOR INTENTION

	Intention	Protocols
Information gathering	Network Infrastructure	ICMP ; SIP ; SNMP ; SSDP
	DNS	DNS ; LLMNR ; MDNS ; NBNS
	ICS	BACnet ; DNP3.0 ; IPMI ; XDMCP ; ...
	Web	HTTP
Attacks	Control	SSH ; SSHv2
	File Sharing	FTP ; TFTP

As IP addresses are mostly dynamic, the same IP address doesn't always belong to the same owner. In the study, we consider that a source is associated with an IP address for 24 hours. At the end of this arbitrary tenure, we consider that the IP address is linked to a new source.

TABLE II. HONEYPOTS STATISTICS

	HP1	HP2	HP3	HP4	HP5
IP addresses	13.677	23.061	25.458	32.229	23.861
Sources	17.644	31.740	22.938	43.887	33.640
Packets	2.847.243	9.678.783	9.434.139	10.299.219	10.706.896
Protocols	41	47	59	48	44
Countries	245	246	241	281	243

Table II shows that the honeypot geolocation has no impact on the cyber attacks they faced. In fact, for all honeypots except HP1 and HP4, the number of packets and the number IP addresses are similar. This phenomena is true for all protocols.The number of different protocols is almost invariant for all honeypots. After analysis, the most prevalent intention is *control*.

In Figure 3 the locations of honeypots are specified as well as the proportion of intentions by the attackers. As shown, all honeypots follow the same pattern with a similar proportion of malicious intentions.

Figure 4 represents the number of messages sent by the 100 most present sources and received by each honeypot. As previously mentioned, the honeypot location has no impact on the data gathered. However, we observe that the intentions of HP1 are significantly below the others.The results obtained for HP1 can be explained by the fact that the server stopped working after 10 days. This was discovered by analysing the PCAP files saved on the server.

Figure 5 represents the number of messages by intention received by each honeypot. As shown, the data obtained for HP3 is also dissimilar to HP2, HP4 and HP5. By analysing

[1]https://dev.maxmind.com/geoip/legacy/geolite

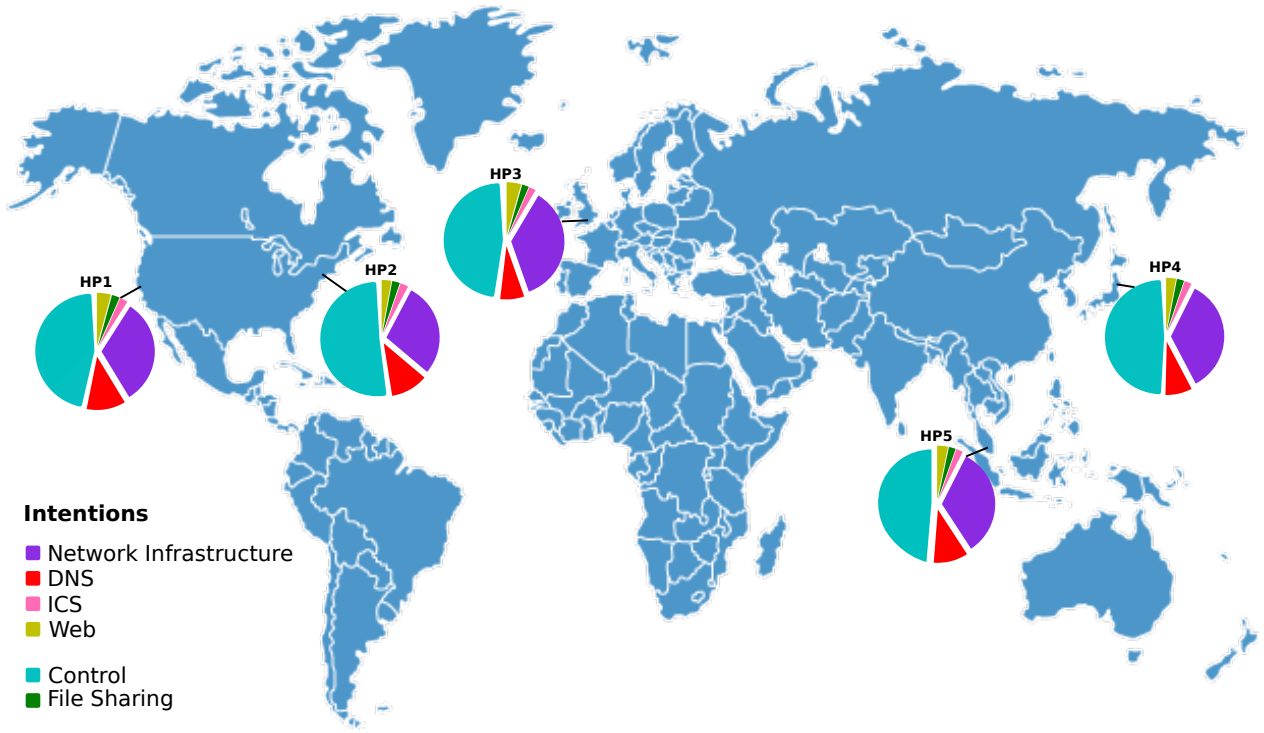

Fig. 3. Honeypots Places and intentions rates

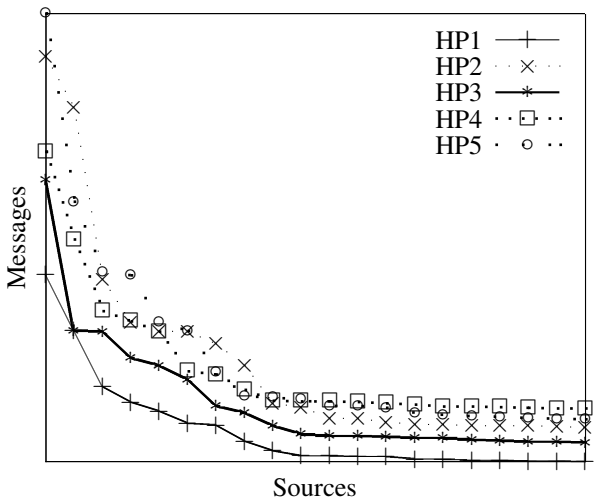

Fig. 4. Number of messages for the 100 most communicating sources

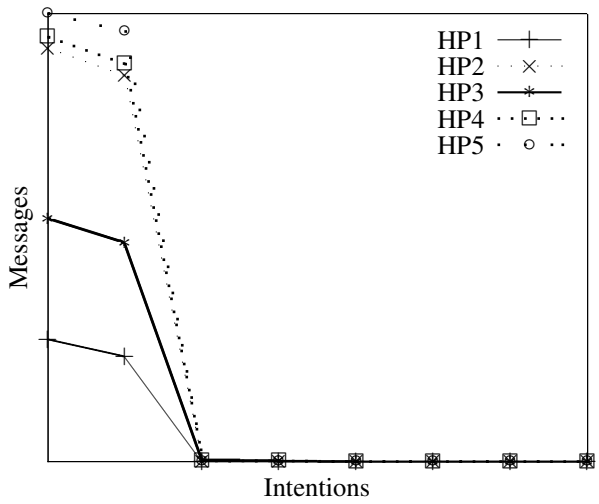

Fig. 5. Number of messages by intention

the intentions of the attacker using the model presented it was also possible to detect an anomaly in the services of HP3 during the acquisition period. Numerous systems and system administrators are unable to consider all the logs due to tremendous amounts generated, moreover, it takes an average of 206 days to detect a cyber-attack on a system [10]. However, through the model presented, and by considering the attacker's intention, it is possible to tremendously reduce the detection of an anomaly in both the services, and the network.

Rule 1 is to reduce the number of elements analysed in the experiment. To this end, different values for the aggregation over time were used: 1 minute, 10 minutes, 1 hour and 1 day.

TABLE III. SIMPLIFICATION PROCESS

	Original data	$1D$	$1H$	$10M$	$1M$
# messages	13 990 758	19 869	51 433	114 365	596 669

Table III shows the differences between the original data and the simplified graph. The simplification process produces a graph with 20 times less messages using 1 minute for time aggregation. This result shows that the graph produced by taking into account intentions reduces the amount of data to analyse while using a small period of time for the aggregation.

V. Conclusion

The exponential amount of data being transmitted over the networks and the increasingly large number of connected devices make it difficult to detect abnormal behaviour. Current research is oriented to the modelling of network flows and detecting anomalies. However, these models and methods often forget human factors and the intention of the attacker. The model presented in this article aims to improve human integration by profiling the human interaction during and after an attack.

This paper also introduces an experiment validation using a set of honeypots spread around the world collected a large amount of raw network data in order to create a dataset for the validation of the graph model presented.

The results showed that the model allows for quick anomaly detection, such as the unavailable services by analysing the attackers intention. Using current tools, it takes an average of 206 days to detect a network breach. The model presented allows to reduce this dramatically by analysing the intention of the attacker. The model can also help identify misconfigurations and service failures as presented in the case study for Honeypot 3.

A simplification process is applied to the model showing that the data to analyse can be reduced by 20 % without losing in accuracy. Furthermore, we plan to study and design new representations based on the model in order to show more hidden patterns. Finally, all the data collected will be freely available online to benefit the scientific community. The method will also be tested on additional datasets extracted from industrial networks and Cyber Physical Systems.

References

[1] Eric Alata. *Observation, caractérisation et modélisation de processus d'attaques sur Internet*. PhD thesis, INSA de Toulouse, 2007.

[2] James F Allen. Towards a general theory of action and time. *Artificial intelligence*, 23(2):123–154, 1984.

[3] Thomas M Chen, Juan Carlos Sanchez-Aarnoutse, and John Buford. Petri net modeling of cyber-physical attacks on smart grid. *IEEE Transactions on Smart Grid*, 2(4):741–749, 2011.

[4] Lawrence E Cohen and Marcus Felson. Social change and crime rate trends: A routine activity approach. *American sociological review*, pages 588–608, 1979.

[5] Allan Cook, Andrew Nicholson, Helge Janicke, Leandros Maglaras, and Richard Smith. Attribution of cyber attacks on industrial control systems. 2016.

[6] Marc Dacier, Van-Hau Pham, and Olivier Thonnard. The wombat attack attribution method: some results. In *International Conference on Information Systems Security*, pages 19–37. Springer, 2009.

[7] John R Goodall. Introduction to visualization for computer security. In *VizSEC 2007*, pages 1–17. Springer, 2008.

[8] Guofei Gu, Roberto Perdisci, Junjie Zhang, Wenke Lee, et al. Botminer: Clustering analysis of network traffic for protocol-and structure-independent botnet detection. In *USENIX security symposium*, volume 5, pages 139–154, 2008.

[9] Clement Guitton. Criminals and cyber attacks: The missing link between attribution and deterrence. *International Journal of Cyber Criminology*, 6(2):1030, 2012.

[10] Ponemon Institute. 2017 Cost of Data Breach Study. Technical report, IBM Security, 01 2017.

[11] Bartosz Jasiul, Marcin Szpyrka, and Joanna Śliwa. Detection and modeling of cyber attacks with petri nets. *Entropy*, 16(12):6602–6623, 2014.

[12] Ines Jguirim, David Brosset, and Christophe Claramunt. Functional and structural analysis of an urban space extended from space syntax. In *8th International Conference on Geographic Information Science, Vienna, Austria*, 2014.

[13] Bin Jiang and Christophe Claramunt. Topological analysis of urban street networks. *Environment and Planning B: Planning and design*, 31(1):151–162, 2004.

[14] Bin Jiang and Ferjan Ormeling. Mapping cyberspace: Visualizing, analysing and exploring virtual worlds. *The Cartographic Journal*, 37(2):117–122, 2000.

[15] Kiran Lakkaraju, William Yurcik, and Adam J Lee. Nvisionip: netflow visualizations of system state for security situational awareness. In *Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security*, pages 65–72. ACM, 2004.

[16] Neal Leavitt. Anonymization technology takes a high profile. *Computer*, 42(11), 2009.

[17] Iyatiti Mokube and Michele Adams. Honeypots: concepts, approaches, and challenges. In *Proceedings of the 45th annual southeast regional conference*, pages 321–326. ACM, 2007.

[18] Andrew Nicholson, Tim Watson, Peter Norris, Alistair Duffy, and Roy Isbell. A taxonomy of technical attribution techniques for cyber attacks. In *European Conference on Information Warfare and Security*, page 188. Academic Conferences International Limited, 2012.

[19] Ludovic Pièrre-Cambacédès and Marc Bouissou. Beyond attack trees: dynamic security modeling with boolean logic driven markov processes (bdmp). In *Dependable Computing Conference (EDCC), 2010 European*, pages 199–208. IEEE, 2010.

[20] Travis C Pratt, Kristy Holtfreter, and Michael D Reisig. Routine online activity and internet fraud targeting: Extending the generality of routine activity theory. *Journal of Research in Crime and Delinquency*, 47(3):267–296, 2010.

[21] Arpan Roy, Dong Seong Kim, and Kishor S Trivedi. Cyber security analysis using attack countermeasure trees. In *Proceedings of the Sixth Annual Workshop on Cyber Security and Information Intelligence Research*, page 28. ACM, 2010.

[22] Christopher J Schreck. Routine activity theory. In *Preventing Crime and Violence*, pages 67–72. Springer, 2017.

[23] Jawwad A Shamsi, Sherali Zeadally, Fareha Sheikh, and Angelyn Flowers. Attribution in cyberspace: techniques and legal implications. *Security and Communication Networks*, 2016.

[24] Lance Spitzner. *Honeypots: tracking hackers*, volume 1. Addison-Wesley Reading, 2003.

[25] W Timothy Strayer, David Lapsely, Robert Walsh, and Carl Livadas. Botnet detection based on network behavior. In *Botnet Detection*, pages 1–24. Springer, 2008.

[26] Meenakshi Syamkumar, Ramakrishnan Durairajan, and Paul Barford. Bigfoot: A geo-based visualization methodology for detecting bgp threats. In *Visualization for Cyber Security (VizSec), 2016 IEEE Symposium on*, pages 1–8. IEEE, 2016.

[27] Roberto Tamassia, Bernardo Palazzi, and Charalampos Papamanthou. Graph drawing for security visualization. In *International Symposium on Graph Drawing*, pages 2–13. Springer, 2008.

[28] Olivier Thonnard, Wim Mees, and Marc Dacier. On a multicriteria clustering approach for attack attribution. *ACM SIGKDD Explorations Newsletter*, 12(1):11–20, 2010.

[29] Jens Tölle, Oliver Niggemann, et al. Supporting intrusion detection by graph clustering and graph drawing. In *Proceedings of Third International Workshop on Recent Advances in Intrusion Detection RAID 2000*, 2000.

[30] M Uma and G Padmavathi. A survey on various cyber attacks and their classification. *IJ Network Security*, 15(5):390–396, 2013.

[31] M Withall, Iain Phillips, and D Parish. Network visualisation: a review. *IET communications*, 1(3):365–372, 2007.

[32] Xiaoxin Yin, William Yurcik, Michael Treaster, Yifan Li, and Kiran Lakkaraju. Visflowconnect: netflow visualizations of link relationships for security situational awareness. In *Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security*, pages 26–34. ACM, 2004.